

Data Management Operating Procedures and Guidelines

The full text of operating procedures, guidelines, and standards, including their underlying *rationale*, and *level of enforcement* are enclosed for your reference.

Numbering schemes follow this format with *nnn* as a sequence number (e.g., 001):

DM OP *nnn* (Operating Procedure)

Operating procedures mandate certain actions that are enforceable for specified IT Project.

DM G *nnn* (guideline)

Guidelines provide best practice methods, and are optional.

Data Management Operating Procedures

DM OP-001 Operating Procedure for Requesting Data Management Services

Rationale:

The Data Services Manager needs certain information to adequately plan and assign data management service resources in support of CMS projects. That information is collected through the *Data Management Service Request Form*, which is designed to capture it in a clear and organized format.

Operating Procedure:

A request for data design services shall be initiated through submission of the *Data Management Service Request Form*.

DM OP-002 Operating Procedure for Identifying System Interfaces

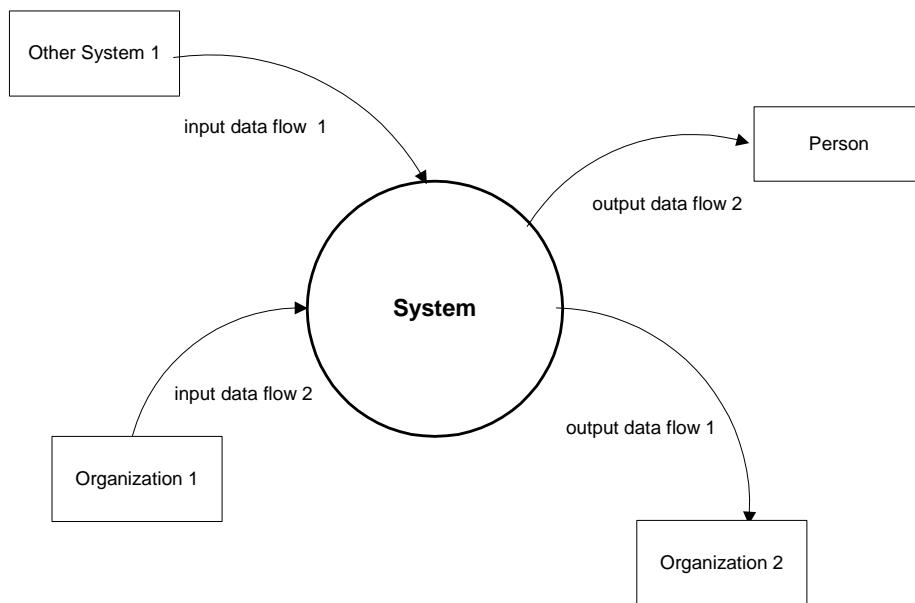
Rationale:

The System Context Diagram concisely illustrates the system boundary in relation to its environment, and documents the actual or planned data flows into and out of the system.

Operating Procedure:

Business application interfaces shall be identified using a System Context Diagram. See example.

System Context Diagram



DM OP-003: Operating Procedure for Developing the Conceptual Data Model

Rationale:

The Conceptual Data Model diagram illustrates the major entities about which the business enterprise needs information. This diagram will assist in the identification of existing data sources and help in determining the need for new entities.

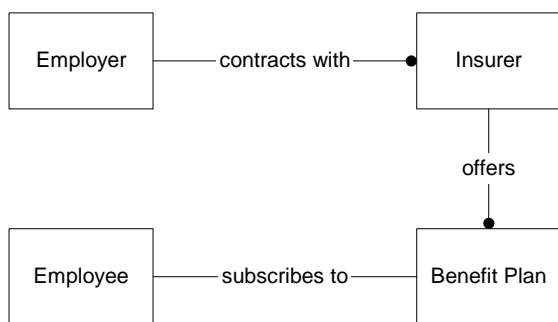
Operating Procedure:

Business applications that create, update, or replicate data shall have high-level data requirements documented in a Conceptual Data Model.

Conceptual Data Models shall represent business entity relationships that are within the scope of the target business function(s) in a manner depicted by the *Example Conceptual Data Model*.

Major IT projects may require Data Architect review

Example Conceptual Data Model



Conceptual Data Model Features:

- Includes the important entities and the relationships among them.
- No attribute is specified.
- No primary key is specified

Business Rule:

An employer contracts with one or more insurance companies to offer health care benefits to any employee who elects to subscribe.

DM OP-004: Operating Procedure for Estimating Data Management Service Needs

Rationale:

The project sizing and estimated cost provides the business enterprise the information for calculation of potential return on investment (*ROI*).

Operating Procedure:

Prior to starting data management services, the Project Sponsor shall ensure development of an estimate of the effort (e.g., dollar amount/hours, data management resources) and schedule of the data services work necessary to satisfy project needs. Data Management services shall be negotiated with the Data Services Manager.

DM OP-005 Operating Procedure for Developing the Logical Data Model

Rationale:

Business requirements are to be documented in a standard notation for consistent documentation with a view for prospective data reuse, and effective communication of business data requirements.

Operating Procedure:

Business applications that create, update, or replicate data shall have their data requirements documented in a Logical Data Model. The Logical Data Model shall be developed in the standard data modeling tool, or in a format that can be directly imported into the standard data modeling tool.

DM OP-006 Operating Procedure for Reuse of Enterprise Entities, Relationships and Attributes

Rationale:

Shared business data artifacts must be managed according to enterprise data stewardship objectives.

Operating Procedure:

Project Data Analysts shall use shared model objects where available rather than duplicating them for exclusive use by each project. Reuse of enterprise entities and attributes requires observance of the following requirements:

1. New data definitions may not redundantly define information already in the Enterprise Metadata Repository. The goal is to ensure that a new data artifact does not duplicate an existing one.
2. When a project selects entity or attributes from the Enterprise Data Model, a log of changes must be kept throughout the project lifecycle. The log must include entries for the following instances:
 - a. Altered entity and attribute definitions
 - b. Changes to cardinality or optionality
 - c. Deletions, migrations, or changes of any kind

DM OP-007 Operating Procedure for Reuse of Enterprise Data Resources

Rationale:

Shared data resources must be managed according to enterprise data management objectives.

Operating Procedure:

Project Data Analysts shall make use of existing enterprise data resources. Reuse of enterprise data resources requires the development of a project Data Source Plan, which identifies candidate sources for the data required to satisfy the project's business requirements. For each candidate data source, the following shall be documented:

1. Name
2. Description
3. Business Owner or steward
4. Project data needs satisfied
5. Limitations (e.g., data quality issues)
6. Constraints (e.g., operational impact of using source)

DM OP-008 Operating Procedure for Defining Data Entities

Rationale:

A well-formed definition makes a data object's contents apparent to business users.

Operating Procedure:

1. An entity shall be defined in a manner that distinguishes its unique role within the business enterprise.
2. The definition shall be clear, concise, and unambiguous. Examples or exclusions may be added to the definition to improve clarity.
3. The definition shall describe a single occurrence of the entity. For example, the description of the entity *Beneficiary* should contain "...an individual that is a designated..." rather than "...individuals that are designated...".
4. Since data entities are not generally technology concepts, the definition may not include references to application privileges, interface behavior or internal database auditing.
5. The definition shall not include references to technology or media. For example references to "tapes" or "disks" are not appropriate in the definition of a business entity.
6. The definition shall exclude references to time-bound or process-bound circumstances. For example, "the information the agency receives quarterly" is not included in a definition.
7. The entity definition may not include acronyms or abbreviations not found in the *Standard and Abbreviations Terms List*.
8. Subject matter experts who understand the business meaning, in conjunction with the data analyst, should develop definitions. If the data analyst develops the definition, it is important that subject matter experts review and confirm the accuracy of the definition.

DM OP-009 Operating Procedure for Naming Data Entities

Rationale:

A good data entity name is meaningful and self-documenting. It provides the first source of information about the entity's purpose and contents.

Operating Procedure:

1. Entity names shall be unique throughout the Enterprise Data Model. This requires that a new entity name not duplicate existing entity names in the Enterprise Data Model.
2. An entity name shall be stated in singular form except in those instances where the plural form is the commonly used term. For example, "savings" when referring to an amount saved.
3. Base the entity name on the respective entity definition.
4. An entity name may contain alphabetic characters A-Z and spaces. It may not contain numbers or special characters, including the slash (/) and hyphen (-).
5. An entity business name has the following structure:

Position	Component	M/O
1	<i>Object Class Term</i> (formerly Prime Word)	Mandatory
2-3	<i>Property Term, Qualifier Term</i> (formerly Modifier Words)	Optional

Example:

Based on the selected terms Equipment (object class term) Source (property or qualifier term), the entity name will be: EQUIPMENT SOURCE .

- ***Object Class Term*** is selected to provide the clearest meaning and alignment with the parent data subject.
 - ***A Property Term*** and one or more ***Qualifier Terms*** are included in the best order for readability and recognition of meaning.
6. Format entity names in mixed case (the first letter of each term is in uppercase, the remaining letters in the term are in lowercase) throughout the model
 7. Avoid using excessively long business names for entities. If a standard business name exceeds the maximum length of 120 characters (including embedded spaces), progressively remove the least significant *qualifier term(s)*.
 8. Use a *space* as the delimiter between terms.
 9. No abbreviations, except when necessary to fit within the 120 character limitation.

10. Use of acronyms in entity names is limited to widely-recognized acronyms that are also designated as *Object Class Terms* or *Property/Qualifier Terms* on the *Standard Terms and Abbreviations List*.

DM OP-009-QR Quick Reference for Naming Data Entities

Quick Reference for Entity Naming Standard:

1. Entity names shall be unique throughout the Enterprise Data Model
2. Entity names shall be stated in singular form.
3. Base the name on the respective entity definition.
4. Exclude special characters.
5. ***Object Class Term*** (mandatory) + most important ***Property Term*** or ***Qualifier Term*** (optional) + [0 to n Qualifier Terms]
6. Format entity names in mixed case (the first letter of each term is in uppercase, the remaining letters in the term are in lowercase) throughout the model.
7. Maximum length of 120 characters
8. Use a *space* as the delimiter between terms.
9. No abbreviations, except when necessary to fit within the 120 character limitation.
10. Use of acronyms in entity names is limited to widely-recognized acronyms that are also designated as *Object Class Terms* or *Property/ Qualifier Terms* on the *Standard Terms and Abbreviations List*.

DM OP-010 Operating Procedure for Defining Data Attributes

Rationale:

A well-formed definition makes an attribute's contents apparent to business users.

Operating Procedure:

1. An attribute shall be defined in a manner that distinguishes its unique role within the business enterprise.
2. An attribute definition shall be clear, concise, and unambiguous. Examples or exclusions may be added to the definition to improve clarity.
3. The definition shall describe a single occurrence of the attribute.
4. The definition shall not include references to technology or media. For example references to "tapes" or "disks" are not appropriate in the definition of a business attribute.
5. The definition shall exclude references to time-bound or process-bound circumstances. For example, "the information the agency receives quarterly" is not included in a definition.
6. The attribute definition may include well-recognized agency acronyms but no other acronyms or abbreviations.
7. Definitions should be developed by subject matter experts who understand the business meaning, in conjunction with the data analyst. If the data analyst develops the definition, it is important that subject matter experts review and confirm the accuracy of the definition.
8. An attribute definition should:
 - a. State the essential meaning of the concept
 - b. Be precise and unambiguous
 - c. Be concise
 - d. Be able to stand alone
 - e. Be expressed without embedding rationale, functional usage, domain information, or procedural information
 - f. Avoid circular reasoning
 - g. Use the same terminology and consistent structure for related definitions

DM OP-011 Operating Procedure for Analyzing Types of Data Attributes

Rationale:

Attributes can be named and defined with additional consideration for how they are used and therefore, enhance their recognition and capability for storing business data values.

Operating Procedure:

Define the attribute in the Logical Data model by the following criteria:

1. **Prime** attributes are basic business facts. They are always added to the model.
2. **Cohesive** attributes are usually processed together for business meaning. They are always added to the model. Use the same object class term and representation term to show the cohesion between the attributes, as in the following recommendation.

Recommended Cohesive Attribute Names			Not Recommended, lacking cohesion		
<i>Object class term</i>	<i>Property/Qualifier term</i>	<i>Representation term</i>	<i>Object class term</i>	<i>Property/Qualifier term</i>	<i>Representation term</i>
Individual	Street	Address	Beneficiary	Street	Address
Individual	City	Address	Individual	City	Location

3. Where possible, use the same names to represent the attributes in all instances, relying on the entity to express the contextual meaning. For example:

Recommended Attribute Names				Not Recommended for Attribute Names			
<i>Entity</i>	<i>Object class term</i>	<i>Property/Qualifier term</i>	<i>Representation term</i>	<i>Entity</i>	<i>Object class term</i>	<i>Property/Qualifier term</i>	<i>Representation term</i>
Beneficiary	Individual	Street	Address	Beneficiary	Beneficiary	Street	Address
	Individual	City	Address		Individual	City	Location
Beneficiary	Individual	Street	Address	Beneficiary	Dependent	Street	Address
Dependent	Individual	City	Address	Dependent	Individual	City	Name

4. **Derived** attributes can be formulated using values from other attributes. They may be added to the Project Logical Data Model if useful to business communication but not necessarily transformed to physical data design. See 220-15 *Modeling Derived Attributes* for additional information about inclusion of derived data in the Project Logical Data Model.
5. **Transaction / Interface transaction elements** are included only if they are to be transformed to the physical data design.
6. **Physical control data**, such as “next record number”, are not included in the logical data model.

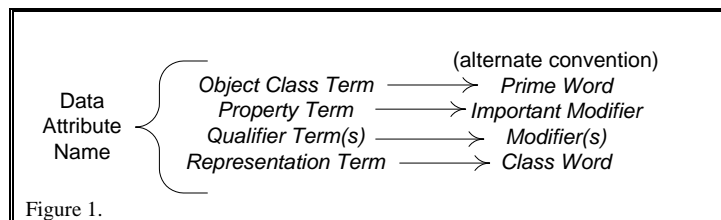
DM OP-012 Operating Procedure for Naming Data Attributes

Rationale:

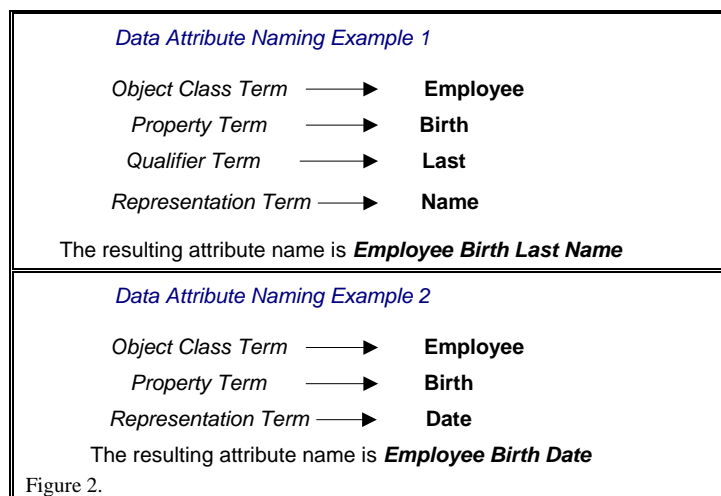
A good data attribute name is meaningful and self-documenting. It provides the first source of information about the data attribute's purpose and contents.

Operating Procedure:

1. A data attribute's name shall be based on its definition. The name suggests the business meaning of the attribute and reflects the most important words in its definition. Match these important words with the terms in the *Standard Terms and Abbreviations List*.
2. A data attribute name consists of: one (1) *object class term*; optionally, a *property term* and/or *qualifier term(s)* (0 or more), and one (1) *representation term*.
3. The CMS data attribute naming convention is largely equivalent to the *prime word/modifier word/class word* convention known by many data analysts. A comparison of the conventions is illustrated in Figure 1.



4. The *object class term* occupies the leftmost (prefix) position in the name.
5. The most significant *property term* and/or *qualifier term* follow the *object class term*.
6. The *representation term* occupies the rightmost (suffix) position.
7. Two examples of how to use the data attribute convention is shown in Figure 2.
8. Format attribute names in mixed case throughout the model



9. An attribute name shall be stated in singular form except in those instances where the plural form is the commonly used term. For example, "savings" when referring to an amount saved.

10. An attribute name may contain alphabetic characters A-Z and the spaces. It may not contain special characters, including the slash (/), hyphen (-), or numbers.
11. An attribute name must not exceed 120 characters. If a standard business name exceeds 120 characters, progressively remove the least significant qualifier term.
12. Use of acronyms in attribute names is limited to widely-recognized acronyms that are also designated as *Object Class Terms* or *Property/Qualifier Terms* on the *Standard Terms and Abbreviations List*.

DM OP-012-QR Quick Reference for Naming Data Attributes

Quick Reference for Attribute Naming Standard:

1. Base the name on the respective attribute definition.
2. A data attribute name consists of: one (1) *object class term*; optionally, a *property term* and/or *qualifier term*(s) (0 or more), and one (1) *representation term*; all terms must appear on the *Standard Terms and Abbreviations List*.
3. The CMS data attribute naming convention is largely equivalent to the *prime word/modifier word/class word* convention known by many data analysts.
4. ***Object Class Term*** occupies the leftmost position of the data attribute name and is selected to provide the clearest meaning and alignment with the parent data entity.
5. ***Property Term and Qualifier Terms*** are included in the best order for readability and recognition of meaning.
6. ***Representation Term*** occupies the rightmost position of the data attribute name.
7. Format attributes names in mixed case throughout the model.
8. An attribute name shall be stated in singular form
9. Space delimiter between terms; no special characters.
10. If a standard business name exceeds 120 characters, progressively remove the least significant qualifier term.

DM OP-013 Operating Procedure for Modeling Derived Data

Rationale:

An example of a useful derived data element is the balance of a bank account. You could derive it from all previous credit and debit statements, since the start balance is 0. However, it would be very unpractical to do this every time the account balance is needed.

Another example of derived data is *totals* and *statistics*, for example the number of *Beneficiaries* in a state.

In the above examples, the data might be stored as part of the record to save processing time. However, one must be careful with storing derived data. When something changes in the source data, the derived data must change accordingly.

If the source data is very dynamic, it's better to just derive it every time.

Operating Procedure:

1. Derived data may be included in the Logical Data Model as a convenient way to communicate the meaning of calculated business data.
2. Derived data may or may not be included in the Physical Data Model, depending on the benefit to actually store the derived product.
3. Use the following criteria in guiding whether to store derived data.

Pro → Store Derived Attribute	Con → Derive Attribute at processing time
Original factors not always available	Original Factors always available
Retrieval performance is important	Updates are frequent
Original factors are static	Original factors are dynamic
Up-to-date accuracy not important	Up-to-date accuracy essential
Derivation is costly or difficult	Derivation is cheap and easy

4. If derived information is included as an attribute, it must be noted as “derived” and the derivation formula must be included in the attribute definition.

DM OP-015 Operating Procedure for Defining Relationships, Cardinality, and Optionality

Rationale:

The labeling of entity relationships in the data model aids the understanding of model details. Also, cardinality labels define the minimum and maximum number of entity occurrences that can participate in a relationship.

Operating Procedure:

1. Redundant relationships between two entities are not allowed.
2. A relationship has only one possible state.
3. Each relationship must be named with a verb or verb phrase to express the relationship in the parent-child direction.
4. Relationship names may be expressed for each direction of the relationship.
5. Non-descriptive relationship names such as “belongs to”, “may have”, “is associated with” or “is a part of” should not be used.
6. Cardinality must be specified for each relationship.
7. Document the description of the entity parent to child relationship to clarify the relationship and describe the conditions of the relationship and any special conditions under which it exists.

DM OP-016 Operating Procedure for Assigning a Primary Identifier

Rationale:

An entity must be defined in a way that the business can distinguish unique occurrences.

Operating Procedure:

1. In the Project Logical Model, every entity type should have a unique identifier composed as one of the following options:
 - a. One attribute
 - b. Multiple attributes
 - c. An identifying relationship and one attribute
 - d. Two identifying relationships
2. An assigned number may be used as an identifier. This is rarely justified but is permitted for practicality when it represents a unique abbreviated way of referring to an occurrence of an entity type or needed to compensate for other unusual circumstances.
3. Examples:
 - a. A generated sequential number may be used to identify a particular entity, where the combination of the identifying attributes and relationship is much too cumbersome for practical use as an identifier.
 - b. During a company merger, records from two sources are to be combined and their existing primary identifiers might result in duplicate identifiers without an assigned number, which acts as a tiebreaker.

DM OP-017 Operating Procedure for Normalizing the Project Logical Data Model

Rationale:

The purpose of data normalization is to ensure that (1) each attribute is assigned to the appropriate entity (2) redundant storage of information is minimized and (3) data structures are stable and receptive to updates, reducing the risk for introduction of anomalies.

Operating Procedure:

1. The Project Logical Model shall be normalized to *First Normal Form*. All attributes of every entity must be atomic, indivisible, and not part of a repeating group.
2. The Project Logical Model shall be normalized to *Second Normal Form*. All attributes of every entity must depend on the full primary key of that entity only.
3. The Project Logical Model shall be normalized to *Third Normal Form (3NF)*. All attributes of every entity must depend on the full primary identifier of that entity only and not be functionally dependent on some other attribute.
4. The Project Logical Model may be normalized to *Fourth Normal Form*. The model is in *3NF* and has no sets of multi-valued dependencies.
5. The Project Logical Model may be normalized to *Fifth Normal Form* such that every join dependency is made through candidate identifiers.

DM OP-018 Operating Procedure for Assigning Domain Value Rules

Rationale:

Value domain assignments support business data definition and quality by restricting data to a certain set of permitted values.

Operating Procedure:

When the business operation depends on data to be within a discrete set of permissible values, the permitted attribute value range must be documented and managed.

DM OP-019 Operating Procedure for Checking Completeness of a Project Logical Data Model

Rationale:

A Project Logical Data Model that conforms to all relevant standards and operating procedures supports the agency's long-term data architecture goals for creation of stable, reusable data artifacts.

Operating Procedure:

Verify that:

1. The entities in the logical model represent all the things of interest within the scope of the business mode.
2. The logical model supports every data need identified in the functional business requirements.
3. The entities and attributes in the logical model have well-formed business definitions.
4. The model adheres to naming standards, notation, and security setting requirements.
5. The logical model is free of data redundancies, incorrect keys, and relationship errors.
6. The logical model is free of derived data or that such data is clearly marked as derived to ensure conscious transference to the physical design layer.
7. The logical model component names are consistent with standard naming conventions.
8. The logical model in is Third Normal Form.
9. The logical model is independent of physical design constraints.
10. Text that identifies the model diagram is located in the upper left corner: the application name; subject area name, if applicable; update date; and modeler identification.

DM OP-020 Operating Procedure for Documenting Data Issues

Rationale:

Sometimes it is not possible to express a data issue in the data modeling tool however, it is still necessary to communicate the issue and track its resolution.

Operating Procedure:

Use the *Data Issues Notes* to report the following:

1. Document business rules that cannot be expressed using the data modeling tool.
2. Document data quality issues of any existing project data that might be in opposition to stated business rules.
3. Document unusual database physical platform location requirements.
4. Document data conversion challenges and any other issues that require special attention from DBA or business users.

Document Data Issues following the example below, including the underlined sections:

Example

Data Issue: The current production table column *BENEFICARY. INDV_WGHT_AMT* has a defined data type of CHAR(3), which has resulted in entry and storage of invalid weight data. Project “*CMIS: Contractor Management Information System*” requires that the column be changed to *smallint*. Therefore existing values that are non compliant with the revision, must be corrected before it can be reloaded to the revised table column.

Date Recorded: 2004/08/24

Resolution: A conversion program has been developed to correct invalid weight data in table column *BENEFICARY. INDV_WGHT_AMT*. Doug Nichols, the project manager will coordinate correction of invalid data with the assigned database administrator, prior to production change implementation.

Date Resolved: 2004/08/25 by *Mable Hanks*, assigned *Project Data Analyst*.

DM OP-021 Operating Procedure for Assigning Information Security Categories

Rationale:

The Federal Information Security Act (FISMA) requires all federal agencies to document the appropriate security categories of asset and operational data.

Operating Procedure:

1. Enterprise data shall be assigned with appropriate security settings in compliance with *FIPS Publication 199 – Standards for Security Categorization of Federal Information and Information Systems*.
2. Each entity in a Logical Data Model shall be assigned a potential impact level for the following security objectives: confidentiality, integrity, and availability.
3. Further guidance is included in the following table.

	POTENTIAL IMPACTS		
Security Objective	LOW	MODERATE	HIGH
<i>Confidentiality</i> Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. [44 USC,SEC.3542]	The authorized disclosure of information could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The authorized disclosure of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals	The authorized disclosure of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals
<i>Integrity</i> Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. [44 USC,SEC.3542]	The unauthorized modification or destruction of information could be expected to have a limited adverse effect on organizational operations, organization assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a serious adverse effect on organizational operations, organization assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organization assets, or individuals.
<i>Availability</i> Ensuring timely and reliable access to and use of information. [44 USC,SEC.3542]	The disruption of access to or use of information or an information system could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

DM OP-022 Operating Procedure for Generating the Project Metadata Repository

Rationale:

The Metadata Repository documents the understanding of business data and facilitates the communication about the information that is represented in the Logical Data Model.

Operating Procedure:

The Metadata Repository holds the following information:

1. For each entity

- a. Entity Name
- b. Entity Definition
- c. Degree of required Confidentiality
- d. Degree of required Integrity
- e. Degree of required Availability

2. For each attribute

- a. Attribute Name
- b. Definition
- c. Primary Key Indicator
- d. Foreign Key Indicator
- e. Data type
- f. Valid Values
- g. Edit Criteria
- h. Business transformation rules
- i. Security category

DM OP-023 Operating Procedure for Performing a Quality Review of the Project Logical Data Model

Rationale:

A review of the logical data model is prudent to ensure its compliance with sound data modeling practices, and CMS Data Management policies and standards.

Operating Procedure:

1. Review the logical model with a Peer Data Analyst that was not assigned to the business project, and if possible the appropriate Data Architect.
2. Review the logical data model for compliance to standards and best practice guidelines. Assess quality of the following areas:
 - a. All business entities are reflected in the logical model.
 - b. Entity names are composed of standard business terms in standard word format.
 - c. Entity names are singular.
 - d. All entities have at least one non-key attribute.
 - e. All entities have at least one relationship.
 - f. All relationships names are verb phrases
 - g. All relationships have specified cardinality.
 - h. There are no redundant relationships
 - i. All entities have well-formed definitions.
 - j. All entities have a primary identifier.
 - k. All attribute names are composed of standard business terms in standard word format.
 - l. All attributes have a well-formed definition.
 - m. All attributes have an assigned class designated data type.
 - n. All attributes follow format standards.

Major IT projects may require Data Architect review

DM OP-024 Operating Procedure for Approval of Data Models based on Business Data Views

Rationale:

Approval by the Business Owner / Partner is required before a Logical Data Model can be considered complete.

Operating Procedure:

Documented approval and signoff by the Project Manager are required to show completion of the Logical Data Model. An email message from the project manager, stating approval is sufficient for this purpose.

DM OP-025 Operating Procedure for Collecting Data Stability and Growth Information

Rationale:

Based on predictions by subject matter experts (SMEs), resources can be better chosen to accommodate near- term entity volume expectations. This procedure encourages collection of data stability and growth information that might be used by the Data Base Manager in selection of the appropriate Data Base Management System (DBMS) platform. (This information serves only as preliminary prediction. Additional, detailed planning for physical storage allocations will be performed by the assigned data base administrator (DBA) in the course of physical database implementation.)

Operating Procedure:

Document the expected growth rate of a new entity predicted by the SMEs, as an annual increase in the number of rows. For example if the SME predicts 100,000 more records in 2 years, the estimated annual growth is 50,000 records.

DM OP-026 Operating Procedure for Completing the Data Models

Rationale:

The data models are stored and cataloged to benefit ongoing enterprise data analysis and business data analysis.

Operating Procedure:

All data models shall be stored and cataloged in DDMSS model management libraries, according to their state of completion.

All models shall be appropriately stored when work is completed (or halted in an incomplete or unapproved status).

DM OP-027 Operating Procedure for Granting Project Data Model Library Access

Rationale:

Storing models in DDMSS libraries provides protected storage for data models and data management documents.

Operating Procedure:

1. Each project shall have designated model management libraries:
 - a. 2 libraries in Model Manager; one for development models, the other for production models.
 - b. 1 folder in the DDMSS Change Control library to hold project Data Issue Documents and Change Control Documents.
2. Full update privileges for development libraries and viewing privileges to project production libraries are available to authorized Project Data Analysts and Data Base Administrators.
3. Update privileges for Model Manager and DDMSS production model libraries are exclusive to the Model Manager Administrator.
4. Assign a name and provide a description for each Model Manager library in the following manner:
5. The library name consists of: *application acronym + application business name + phase designation + approval status*.
 - a. Example: CMIS: Contractor Management Information System Approved.
6. The application library description should be brief and may include the names of the assigned data analysts, project manager, and government task leaders (GTL).
 - a. Example: *As an intended replacement for the CROWD system, CMIS allows CMS to effectively manage, monitor, and report on the performance of its Medicare fee-for-service contractors. Project Manager is Doug Nichols of Acme Systems, Inc. The Project Data Analyst is Mable Hanks. The Project Database Administrator is Nancy Owens.*
7. Create a folder in the *DDMSS Change Control Library*.
8. The folder name convention is: *application acronym + application business name + phase designation + YYYYMMDD*.
 - a. Example: CMIS: Contractor Management Information System 20030401.

DM OP-028 Operating Procedure for Naming and Defining Data Models

Rationale:

Consistent names and description formats are important to the cataloging of data models, managing them as data architecture resources, and exploiting their potential for reuse.

Operating Procedure:

A data model is to be named in the following manner:

system acronym + (“relational”/”dimensional”)+ model type (EDM/CDM/LDM/PDM) + approval date (or the storage date for models in development) in yyymmdd format.

Example: **CMIS relational LDM 20040726.**

The model definition describes the purpose and status of the model in a few sentences of text. It is optional.

Example: **Current approved LDM.**

DM OP-029 Operating Procedure for Selecting Model Type

Rationale:

The CMS standard data modeling tool, *All-Fusion ERwin Data Modeler*, provides convenient functions that help the data modeler to create models that support the various levels of CMS data architecture.

Operating Procedure:

Select the model type according to project needs. Logical models capture data requirements. Physical models support the development of a database design in parallel with or separately from the associated conceptual or logical model development.

1. The following table lists the allowed model source relationships for logical and physical models:

Parent Model Type	Child Model Type	Notes
Logical	Logical	For example, enterprise to project logical, or project conceptual to project logical
Logical	Physical	You can derive a physical model from a logical model. The result is a pair of linked models with the logical model as the parent and the physical model as the child
Physical	Physical	For example, generic physical to target server specific physical

2. A third model type, logical/physical, created by ERwin version 3.5.2 or earlier, combines one logical model and one physical model that are tightly coupled. Use a logical/physical model when you require only one physical implementation for a given logical model and do not need to continue logical modeling while the physical design is being implemented. Logical/physical models can also be created by reverse engineering an implemented database or a DDL script. Logical/physical models are used in design layer architectures, but their usage is a little different. The allowed model source relationships that result from splitting a logical/physical model are:

Parent Model Type	Child Model Type	Notes
Logical/Physical	Logical	The logical level and model properties such as the naming standards model and data type standards model may be passed to the child model
Logical/Physical	Physical	The physical level, and model properties such as the naming standards model and data standards model may be passed to the child model

Data Management Guidelines

DM G-001 Guideline for Assigning Data Analysts

Rationale:

The quality of project data analysis and design deliverables relies on the capabilities of the Project Data Analyst.

Guideline:

Assign Project Data Analysts that have practical experience that is commensurate with the project's data analysis needs and the anticipated design complexity.

Suggested Skill Set:

1. A minimum of five years experience in Relational and/or Dimensional Modeling techniques.
2. Skills for compiling business requirements analysis and documentation.
3. Experience interfacing with health care professionals and business customers.
4. Skills and experiences with Medicare subject matter domains.
5. Knowledge of the health care insurance industry.
6. Experience with standard CMS data modeling tools.

DM G-002 Guideline for Conducting the Logical Data Design Kickoff Meeting

Rationale:

The purpose of this guideline is to encourage an understanding of participant roles and responsibilities, discuss major project goals, and confirm the level of agency sponsorship including project budget authorization. The meeting also provides an opportunity to discuss the data design process, the checkpoint reviews, and the location and scope of the data administration standards.

Guideline:

Follow the outlined agenda and exchange information as indicated.

Agenda

1. Introductions of team members and roles
2. Explain data design process
3. Identify level of service needed and available funding sources
4. Identify schedule requirements
5. Collect data requirements documents from Project Manager
6. Estimate time frame for follow-up response

DM G-003 Guideline for Monitoring Schedule and Cost of Data Services

Rationale:

Monitoring of logical data design costs is necessary to avoid unapproved costs and schedule overruns.

Guideline:

Monitor baseline estimates and report compliance on a periodic basis (e.g., bi-weekly) to the Project Manager.

DM G-004 Guideline for Designating Representation Term and Data Type

Rationale:

Attributes can be named and defined with additional consideration for how they are used and therefore, enhance their recognition and capability for storing business data values.

Guideline:

1. Suffix an attribute name with the appropriate *Representation Term* that fits with the anticipated category of data values to be stored.
2. There may be several physical data types that could be used to represent business data values for a domain. For example an integer data type (INTEGER, SMALLINT, TINYINT) could be used to store a domain whose valid values are integers between 1 and 20. Alternatively, a decimal data type could be used.
3. In selecting the data type, consider the manner in which the RDBMS of choice stores values and the application processing efficiency.
4. The Representation Term and data type must not conflict; for example an attribute ending with the representation term *Description* must not have a data type of *Decimal* assigned.
5. More examples of compatibility between Representation Term and data type are shown in the table below.

CMS Standard Representation Terms and examples of associated Data Types			
TERM	ABBREVIATION	DEFINITION	Example data types
ADDRESS	ADR	Alphanumeric data which identifies street name and number, suite number, etc. May also be used for Post Office box, email address, or Internet address (URL).	String, Varchar, Char(50) Decimal (6,2)
AMOUNT	AMT	A number that is expressed as a currency denomination.	Currency
AUDIO	AUD	A type of Object that is capable of storing a stream of audible sounds.	String, Varchar
CODE	CD	A short character string used to replace a more descriptive value.	String, Char (50)
DATE	DT	A calendar date (month, day, and year) documenting when something happened.	Numeric
MESSAGE	MSG	A brief explanation or warning of a condition such as an edit validation error.	String, Varchar
QUANTITY	QTY	A type of numeric value that is a non-monetary measure of a dimensional extent, volume, etc.	Numeric, Integer
TIMESTAMP	TS	Date and time combined, documenting when something is recorded by a system.	Datetime

DM G-005 Guideline for Constructing Physical Data Names

Rationale:

Physical data names represent the technical implementation of a fact that is important to the business. Therefore, it is important that the names of physical data can be easily recognized and associated with the business concept.

Guideline:

1. Physical data name lengths must comply with the name length limitation of the target RDBMS.
2. Format physical names in uppercase.
3. Replace each term in the *Attribute Name* with the abbreviated standard term. The standard abbreviations are listed in the *Standard Term Report*.
4. Once the abbreviations are applied, if the abbreviated data name exceeds 30 characters (or the physical database name limit), remove the least important *Qualifier Term* or *Property Term* abbreviation, one at a time, until the name fits the name length limitation.
5. Replace the spaces between the terms with underscores.
6. Approved acronyms can be used in place of standard abbreviations. For example, *CTR_MDCR_MD CD_SRVC* can be represented by *CMS*. (Only approved acronyms are allowed.)
7. Underscores between *Property Terms* and *Qualifier Terms* may also be removed if the name needs to be further shortened to fit within the size limitation.
8. Removal of the *Object Class Term* abbreviation is prohibited.
9. Removal of the *Representation Term* is prohibited.
10. Examples:

Logical Attribute Name	Physical Column Name
ORGANIZATION ENROLLMENT DATE	ORG_ENRL_DT
INDIVIDUAL LAST NAME	INDV_LST_NME
PROVIDER INSURANCE DENTAL COVERAGE CODE	PRVDR_INSR_DNTL_CVRG_CD

DM G-006 Standard for Assigning Date Formats

Rationale:

For date and time attributes, Data Administration has adopted the International Standard for the representation of dates and times, ISO 8601. Its full reference number is ISO 8601: 2000 (E), and its title is "Data elements and interchange formats - Information interchange - Representation of dates and times." ISO 8601 describes a large number of date/time formats. The standard is based on the 24-hour timekeeping system. To reduce complexity, only the following date/time formats are being supported:

Guideline:

Content Represented	Format	Example
Complete date	YYYY-MM-DD	2003-10-08
Year and month	YYYY-MM	2003-10
Year	YYYY	2003
Complete time	hh:mm:ss	18:33:56
Hour and minute	hh:mm	18:33
Hour	hh	18

DA has approved these formats for logical design; this standard is NOT intended to constrain physical implementation.

DM G-007 Guideline for Conducting the Data Architecture Review

Rationale:

The purpose of this guideline is to obtain an agreement on the sources of project entity level data as early in the data design process as possible. Ideally, an early agreement will give all the impacted parties the time to accommodate the new data needs.

Guideline:

Follow the outlined agenda and exchange information as indicated.

Required Materials:

1. A draft Project Logical Data Model, which, at a minimum, shows the project entities. (Detailed level attributes are not required for the Architecture Review.)
2. A Data Source Plan
3. An Impact Analysis Report

Agenda:

1. Introduce the project principals and briefly discuss the project's purpose.
2. Based on the Data Source Plan:
 - a. Review the EDM candidate entities that are being proposed for reuse;
 - b. Review the non-EDM data sources that are being proposed for reuse; and
 - c. Discuss / justify the need for any new entities.
3. Based on the Impact Analysis Report:
 - a. Discuss the pros and cons for production application systems that might be impacted by the project application's shared use of existing data structures.
4. Request timely approval of the Data Source Plan (or an alternative recommendation) from the responsible Data Architect(s) and Data Steward(s).

DM G-008 Guideline for Mapping Data Requirement Sources

Rationale:

Mapping is performed to validate attribute-level completeness of the Project Logical Data Model to meet business needs, to assure that attributes have a source of their values defined and to confirm any requirements for aggregated or derived data..

Each attribute in a Project Logical Data Model must contribute to at least one business requirement. Requirement statements in the standard requirements management tool are cross-referenced to each attribute in the data modeling tool.

The origin of an attribute's content can be:

- User input
- Database column(s) and containing table or view
- Data exchange standard, message type and field(s)
- Software process

This information may be documented from an initial load perspective, and from an on-going operations perspective.

Guideline:

Record the corresponding requirement information for each attribute using the Attribute UDP named Attribute Requirement ID

Record the data source information for each attribute using the Attribute UDP named Attribute Data Source Name.

If applicable, document the transformation algorithm used to compute the attribute's value in the Attribute UDP named Attribute Derivation Text.

The Project Data Analysts' role in developing, or supporting documentation of this information may vary.

See [Data Modeling tool standard for Creating Project Logical Data Models](#) for tool specific details.

DM G-009 Guideline for Preparing the Project Logical Data Model for Physical Design

Rationale:

The logical data modeler and physical data modeler may or may not be the same individual. In either case, the logical data modeler should anticipate what the physical data modeler needs to know in order to develop an efficient physical data model and capture these suggestions as comments in the entity definition

Guideline:

1. Consider the performance of record retrieval based on anticipated record volume and the caching, indexing, and clustering facilities available in the anticipated DBMS.
2. Consider the mandated enterprise data security settings and any conflicts with the proposed business data functional requirements. For example, whether data will be distributed to multiple platforms or outside of firewalls.
3. Make notes of additional concerns and insights, which can later be considered in physical data design.
4. Do not modify the project logical data model to reflect physical design compromises such as denormalization to roll down supertype attributes into subtype entities. Information pertinent to physical design considerations are to be recorded as comments.
5. Provide the first-cut physical names using the approved abbreviated terms and construction rules defined in *DM G-005 Guideline for Constructing Physical Data Names*.

DM G-010 Guideline for Constructing Physical Table or Files Names

Rationale:

Physical data names represent the technical implementation of something that is important to the business. Therefore, it is important that the names of physical data can be easily recognized and associated with the business concept.

Guideline:

1. Physical data name lengths must comply with the name length limitation of the target database.
2. Format physical names in uppercase.
3. To create the physical table/file name, replace each term in the logical *Entity Name* with the standard term abbreviation. The standard abbreviations are listed in the *Standard Terms and Abbreviations List*.
4. Once the abbreviations are applied, if the abbreviated data name exceeds 30 characters (or the physical database name limit), remove the least important *Qualifier Term* or *Property Term* abbreviation, one at a time, until the name fits the physical name length limitation.
5. Replace the spaces between the terms with underscores.
6. Approved acronyms can be used in place of standard abbreviations. For example, *CTR_MDCR_SRV* can be represented by *CMS*. (Only approved acronyms are allowed.)
7. Underscores between *Property Terms* and *Qualifier Terms* may also be removed if the name needs to be further shortened to fit within the size limitation.
8. Removal of the *Object Class Term* abbreviation is not allowed.
9. Examples:

Logical Entity Name	Physical Table Name
ORGANIZATION ENROLLMENT	ORG_ENRL
ORGANIZATION	ORG
PROVIDER INSURE DENTAL COVERAGE	PRVDR_INSR_DNTL_CVRG

DM G-011 Guideline for Constructing Physical Column or Element Names

Rationale:

Physical data names represent the technical implementation of a fact that is important to the business. Therefore, it is important that the names of physical data can be easily recognized and associated with the business concept.

Guideline:

11. Physical data name lengths must comply with the name length limitation of the target RDBMS.
12. Format physical names in uppercase.
13. Replace each term in the *Attribute Name* with the abbreviated standard term. The standard abbreviations are listed in the *Standard Term Report*.
14. Once the abbreviations are applied, if the abbreviated data name exceeds 30 characters (or the physical database name limit), remove the least important *Qualifier Term* or *Property Term* abbreviation, one at a time, until the name fits the name length limitation.
15. Replace the spaces between the terms with underscores.
16. Approved acronyms can be used in place of standard abbreviations. For example, *CTR_MDCR_MDCD_SRVC* can be represented by *CMS*. (Only approved acronyms are allowed.)
17. Underscores between *Property Terms* and *Qualifier Terms* may also be removed if the name needs to be further shortened to fit within the size limitation.
18. Removal of the *Object Class Term* abbreviation is prohibited.
19. Removal of the *Representation Term* is prohibited.
20. Examples:

Logical Attribute Name	Physical Column Name
ORGANIZATION ENROLLMENT DATE	ORG_ENRL_DT
INDIVIDUAL LAST NAME	INDV_LST_NME
PROVIDER INSURANCE DENTAL COVERAGE CODE	PRVDR_INSR_DNTL_CVRG_CD

DM G-012 Guidelines for Assigning Data Planning Participants

Rationale:

Business management and IT management must work together to create an Enterprise Architecture (EA) for CMS business functions.

Guideline:

Assign CMS personnel and responsibilities as described below.

1. CMS Enterprise Architect (FEA liaison)
Maintains the enterprise architecture to ensure that business processes and information systems are aligned.
2. Enterprise Data Architect
Identifies data design and integration issues.
3. Enterprise Process Architect
Identifies process design and integration issues.
4. Business Owners / Partners
Provide business information requirements and security category specifications.
5. Data Stewards
Ensure that all processes enforce data quality.

DM G-013 Guidelines for Conducting Enterprise Data work sessions

Rationale:

This is a guideline for conducting Information Technology Systems Planning work sessions related to Enterprise Data Architecture.

Guideline:

Follow the outlined agenda and exchange information as indicated.

1. Schedule design sessions with Business Intelligence Council and meeting facility.
2. Conduct orientation and training for design session participants.
3. Prepare the materials, room, and software aids.
4. Customize the design session agenda.
5. Conduct the kickoff meeting.
6. Agenda
 - a) Introductions of BIC team members and roles
 - b) Designate the BIC Chairperson.
 - c) Establish the need for alignment between business functions with data architecture.
 - d) Select team members for subsequent detailed definition component and break-out sessions.
 - e) Explain how the EA project fits in the strategic direction of the agency.
 - f) Identify level of service needed and available funding sources
 - g) Explain the definition design process:
 - h) Confirm the general requirements for the business functions areas affected, including the scope, goals and objectives.
 - i) List the main processes within each affected business function area.
 - j) Develop blueprints for and descriptions of the process and enterprise entities in use (the as-is).
 - k) Identify the information needed for supporting the processes and depict the enterprise entities in a logical data model (the to-be).
 - l) Document expected individual contributions.
 - m) Identify schedule requirements.
 - n) Estimate the timeframes for follow-up responses.
 - o) Publish work session minutes.

DM G-014 Guideline for Creating a Business Process Diagram

Rationale:

This is a guideline for development of a Business Process Diagram.

Guideline:

1. Define the process scope.
 - a. Describe what is being done.
 - b. Tie the process to an organization and business application.
2. Set the level of detail
 - a. Scope the diagram to represent a major business process at *Level 0*. (This is equivalent to a FEA-BRM chartered function such as *Maintain Medicare Beneficiary Enrollment*).
 - b. Decompose the *Level 0* process to 8-10 *Level 1* sub-processes.
 - c. If needed to show all business recognized tasks, decompose respective *Level 1* sub-processes to *Level 2* sub-processes.
3. Present the processes in graphical format.
 - a. It is not necessary to show sequence.
 - b. Is not necessary to express conditions that initiate a process.
 - c. Show organization interaction (who does what).
 - d. Show information exchanges (what is needed to do this).
 - e. Identify the FEA- BRM alignment.

DM G-015 Guidelines for Defining a Subject Area

Rationale:

Data architecture is more helpful to business personnel when it is organized in a manner that speaks to business purposes. That is the objective of a Subject Area: to serve as the broadest classification for a business line's data.

Guideline:

Create a Subject Area by the following criteria:

1. Name and define a Subject Area according to business line. Use a work session to allow business users to discuss their processes and come up with an overall name for a grouping of enterprise entities. Good subject area names usually reflect a major business grouping of data. Occasionally, a subject area name might be the name of a particular business activity that an organization keeps information about, for instance *Beneficiary Enrollment*.
2. Enterprise Entities are exclusive to a single Subject Area. Do not duplicate enterprise entity or subject area names.
3. All *Enterprise Entities* must have an assigned *Subject Area*.
4. Data Stewards should be stakeholders from all parts of the Subject Area business line. Try to assign Data Stewards from the organization unit that is as close as possible to the initial point of data capture of the most important Enterprise Entities in the *Subject Area*.
5. If a business change means that certain business functions end, determine whether Enterprise Entities can be removed from Subject Area(s) and the Enterprise Data Model.

DM G-016A Guidelines for Analyzing and Approving Standard Terms

Rationale:

A consistent use of business terms leads to a common vocabulary and understanding of business data. Therefore, it is important that the business taxonomy be managed and used appropriately to label data entities and attributes.

Guideline:

If terms in the *Standard Term List* are inadequate for naming a data entity or attribute, a request may be made to add a new standard term to the list.

New terms are analyzed according to the following criteria:

1. All standard terms are to be atomic, containing no special characters e.g., hyphens, slashes, or underscores.
2. Terms are accepted in singular form e.g. “Vendor” is acceptable “Vendors” is not.
3. Slang or buzz words are not acceptable e.g. “Zippy”, or “Buy-in”.
4. Terms are accepted in the present tense e.g., “Interviewed” is accepted as “Interview”.
5. Terms are accepted in their root form e.g., the acceptable form of “Rotation” is “Rotate”.
6. When a word can be expressed in different forms to indicate possession or direction, multiple forms of a term can be accepted e.g., “Payee”, “Payer”, and “Payment”.
7. Connectors are not accepted e.g., “Of”, “By”, “Per”, “Or”, “And”.
8. Avoid addition of synonyms to the list, except in those instances where the synonym’s use is justified for business communication. For example, when the term “Responsibility” is an existing standard term, resist adding the synonym “Duty”.
9. Avoid negative terms such as “Non-participant”, “Unqualified”, “Ineligible”, etc. Rationale: the use of negative terms invariably leads to confusing names; also, they have the potential for creating double negative conditions.

Term approval criteria	Example of data name, using negative term - “Ineligible” versus positive term - “Eligible”	Example of data value	Likely business interpretations or questions
Avoid negative terms	Person Ineligibility Switch	“N”	The “N” value means that the person is eligible, or does it?
	Person Ineligibility Switch	“Y”	Does this mean the person is eligible?
Allow positive terms	Person Eligibility Switch	“N”	Clearly, the person is not eligible.
	Person Eligibility Switch	“Y”	Clearly, the person is eligible.

10. Explicit criteria for Object Class Terms (compliance with ISO/IEC 11179 – 5): An Object Class Term is a noun that represents an important entity classes or entity subclasses in an enterprise data subject area. Examples of Object Class Terms: Account, Agreement, Budget, Employee, Location, Procurement, Supplier
11. Explicit criteria for Property Terms (compliance with ISO/IEC 11179 – 5): A Property Term expresses a discrete category. Examples of Property Terms: City, Gender, Spouse, Report, Revenue

Explicit criteria for Qualifier Terms (compliance with ISO/IEC 11179 – 5): A Qualifier Term can be used to further modify or specify an Object Class Term or Property Term. Examples of Qualifier Terms: Minimum, Net, Last, Temporary

12. Explicit criteria for Representation Terms (compliance with ISO/IEC 11179 – 5).
13. Representation Term describes the valid form of data content. Add new terms only when a new form of data can not be accommodated or expressed by the datatypes associated with existing Representation Terms. (See DM G-004 Standard for Designating Representation Term and Data Type)
14. Examples of Representation Terms: Code, Identifier, Number, Date.

DM G-016B Guidelines for Abbreviating a Standard Term

Rationale:

Consistent abbreviations of business terms support the naming of physical data artifacts. Therefore, it is important that an abbreviation of a term can be easily recognized and readily associated with the original business term.

Guideline:

If there is no common or industry-wide abbreviation for a word, use the following guidelines to create an abbreviation. (Example abbreviated terms are shown in brackets.)

1. Words that are four letters or less are not abbreviated unless there is a widely-recognized and common abbreviation available.

Example: AGE [AGE]

There are exceptions such as: DATE [DT], CODE [CD]

2. Condense all double constants to a single constant.

Example: Enrollment [ENRL]

3. Eliminate vowels except: a.) when it is a leading vowel; b.) when the vowel is needed for clarity; c.) when the vowel is needed to provide uniqueness; d.) when a vowel is part of a commonly used abbreviation.

Example of a: Employee [EMPY]

Example of b: Exposition [EXPO]

Example of c: Library [LIB] and Liability [LIAB] (An “A” is added to distinguish Liability’s abbreviation).

Example d: Product [PROD]

4. Abbreviations are limited to eight characters. Abbreviations of 4 characters are preferred.

Discouraged example: RESPONSIBILITY [RSPNSBLTY]

Suggested example: RESPONSIBILITY [RESP]

5. Abbreviations must be unique in order to support clarity of usage.

First term example: RESPONSIBILITY [RESP]

Second term example: RESPONSE [RSPN]

6. An abbreviation may not form a word. If the abbreviation forms a word, replace letters to make the abbreviation not be a word while maintaining uniqueness.

Prohibited example: RETENTION [RENT]

Suggested example: RETENTION [RTNT]

7. Abbreviated words that are widely known and consistently used in the agency should be defined as an acronym.

Example: Health Service Area [HSA]

8. Numbers are not abbreviated, and can be attached to a word.

First example: 1993 [1993]

Second example: ALTERNATIVE 1 [ALTN1]

DM G-017 Guidelines for Reference Tables

Rationale:

The purpose of a *reference table* is: 1) to categorize, classify, qualify, and constrain other data; 2) to provide a method of coding and decoding; and 3) to ensure that only valid values are entered into a database.

Guideline:

1. Reference tables must be free of defects.
2. Reference tables are part of physical data design.
3. Reference tables may be included on the logical data model at the discretion of the project data modeler and the data architect.

The use of reference tables is favored under the following circumstances:

4. If the number of allowed values is at least twenty (20);
5. If the set of allowed values is subject to change or grow; and
6. When it is simpler to change values in a table rather than updating program logic or adding a physical check constraint.

Reference tables tend to follow a standard format:

7. *Primary identifier, Code, Full Name, and a Description.*

DM G-018 Guidelines for Managing Data Values through Physical Constraints

Rationale:

The integrity of data values in data resources is important in providing an organization with quality information. Database Management Systems provide various methods that can automatically check and ensure that values comply within an established range.

Guideline:

The following list describes types of constraints.

1. *Primary Key Constraint* - The *primary key* of a table holds a value that uniquely identifies each row in the table. It is defined when a table is created or altered. A table may contain only one *primary key*.
2. *Foreign Key Constraint* - A *foreign key* is a column or combination of columns that form a relationship between the data in two tables. The relationship is created between two tables by adding the column or columns that hold one table's primary key values to like column or columns in the other table. The second table holds the *foreign key*.
3. *Unique Constraint* - A *unique* constraint ensures that no duplicate values are entered in specific columns that do not participate in a *primary key*.
4. *Check Constraint* - A *check* constraint enforces domain value integrity by limiting the values that are permitted in a column. Like *foreign key* constraints, *check* constraints control the values that are placed in a column. In contrast, *check* constraints limit the valid values to a *declared* list of values.
5. *Default Constraint* - The *default* constraint determines the value to be stored when no value is explicitly assigned.
6. *Null Constraint* - This determines whether a column accepts a NULL value.

Integrity constraints attempt to enforce data consistency and prevent accidental damage to the database during updates. Use the table below as a guideline for when to apply alternative constraint types.

Managerial Circumstance	Constraint Type			
	Assertions	Check Constraints	Foreign Key (FK) Constraints	Triggers
A stable set of permitted values		√		
Occasional change to the set of permitted values		√	√	
The data value is dependent on variable conditions.	√			√

DM G-019 Guideline for Modeling Supertypes and Subtypes

Rationale:

It may be helpful to partition an entity type into subtype entities when: it helps business communication to show a subtype entity with a more specialized meaning; or when a subtype entity requires different integrity treatment.

Guideline:

1. A supertype or subtype is named according to entity naming standards.
2. A subtype has at least one attribute or one relationship that is not in the supertype.
3. Sometimes it is simpler to combine a supertype and subtype within an entity. However when subtypes are represented within a single entity, the attributes specific to the subtype are defined as *optional* (nulls allowed). This permits the entity to exist when the subtype does not, as illustrated in the left-side example in the following table.

Example: combined Supertype /Subtype	Example: partitioned Supertype / Subtype
<i>Employee</i> <i>Employee Number(required)</i> <i>Employee Name(required)</i> <i>Employee Job Code(required)</i> <i>Employee Street Address(required)</i> <i>Employee Temporary Type Code(required)</i> <i>Employee Temporary Company Name (optional)</i>	<i>Employee</i> <i>Employee Number(required)</i> <i>Employee Name(required)</i> <i>Employee Job Code(required)</i> <i>Employee Permanent Address (required)</i> <i>Employee Temporary</i> <i>Employee Number(required)</i> <i>Employee Temporary Company Name (required)</i>

4. Note the attribute *Employee Temporary Type Code*; this attribute distinguishes whether the record holds data for a temporary employee or not and is referred to as a *type code* or a *discriminator*. *Discriminators* are needed when subtypes are generalized into the supertype.
5. Balance the need for subtyping by looking at business rules: create a subtype when there is a distinct need for subtype *referential integrity*; otherwise, generalize the subtype within the supertype.
6. Observe normalization rules in order that the Logical Data Model transforms appropriately into a Physical Data Model.

DM G-020 Guideline for Using ERwin Complete Compare to Create a Brand New Data Model

Rationale:

To manage data models as an enterprise information resource, it is necessary for the model metadata to be collected in a consistent form. The Tool Use standard defines how standard UDPs will be used as the mechanism to promote such consistency. This guideline describes the procedure for incorporating the standard data model UDPs into a brand new model.

Guideline:

When a modeler is tasked with creating a brand-new data model, the templates can be used as a start point to facilitate the introduction of the standard UDPs. The modeler will simply:

1. Open up the relevant logical or physical data model template
2. Do a “save as” in order to save the template under the name desired for the brand-new data model.
3. The modeler can now open up the model just created from the template. This model will initially contain absolutely no entities, tables, etc. But as a consequence of having used the template as a start point, every new object that is subsequently added (manually or by using "Complete Compare") to the new model will immediately have the correct standard UDPs, ready for whatever entry is relevant. The modeler is thus spared the time-consuming effort of having to add the UDPs to new data model.

DM G-021 Guideline for Using ERwin Complete Compare to Import Standard Logical Model UDPs

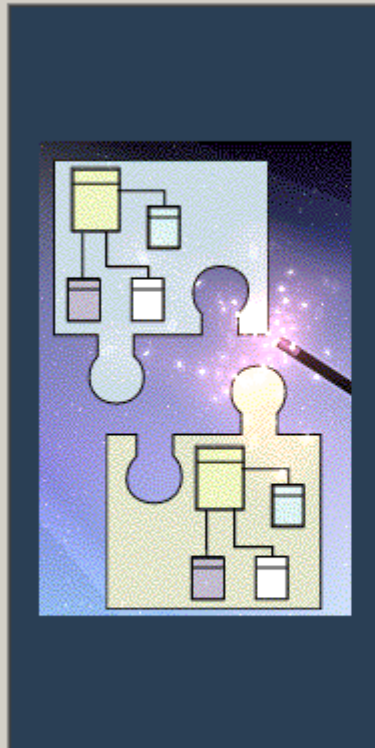
Rationale:

To manage data models as an enterprise information resource, it is necessary for the model metadata to be collected in a consistent form. The Tool Use standard defines how standard UDPs will be used as the mechanism to promote such consistency. This guideline prescribes the procedure for incorporating the standard logical data model UDPs into an existing model.

Guideline:

- A. It is imperative that you first **make a copy** of the ERwin logical data model that you wish to import UDPs into. Then immediately **rename** this copy to clearly indicate that it is the target logical data model that will be used in the import attempt. This will preserve the original logical data model in case something unwanted happens to the target logical data model during the Complete Compare work.
- B. Open up the ERwin target logical data model. Since we are talking about importing logical model UDPs, this target model will either be an ERwin “pure” logical data model, or the logical view of an ERwin logical\physical data model.
- C. From the ERwin toolbar go to “Tools\Complete Compare” and:
 - a. Set “Compare Type” to be “Model level compare”.
 - b. For the “Compare Current Model with” use the Browse feature to locate the path and file where you have saved the logical model UDP template, named “Std_LDM_UDP_Template”.
 - c. Set “Sync Action” to be “Update Current”.
 - d. Click “Next”.

Update Model - Set Options



Compare Options

Compare Type

- ☐ Database level compare
☒ Model level compare

Compare Current Model with

- ☐ Database ☐ ModelMart
☐ Script File ☒ ER1 File

Browse...

C:\Chastain Documents\Clark Erwin Tests\Std_LD

Sync Action

- ☐ Bidirectional
☒ Update Current
☐ Update Other

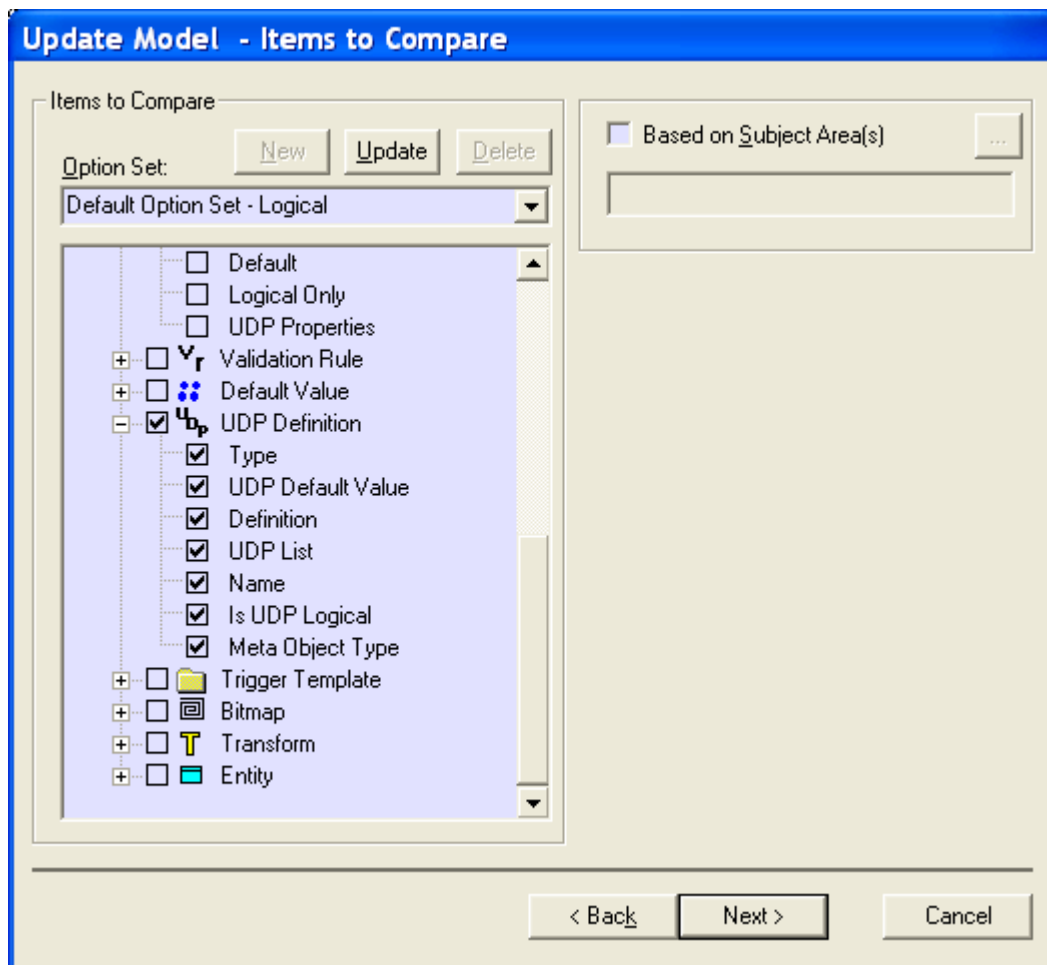
< Back

Next >

Cancel

- D. At the “Items to Compare” screen deliberately unselect everything except the UDP Definition item. All the UDP Definition subordinate items (Type, UDP Default Value and so on) definitely need to remain selected (with a checkmark next to each one). Click “Next”.

(Note that you won't be able to uncheck the Domain item – ERwin keeps that item selected. To prevent Domains from being compared, uncheck all the items subordinate to the Domain object type – Domain Name, Domain Icon and so on.)



Update Model - Items to Compare

Items to Compare





Option Set:

New

Update

Delete

Default Option Set - Logical

- ☐  Default Font & Color
- ☒  Domain
 - ☐ Domain Name
 - ☐ Domain Icon
 - ☐ Domain Definition
 - ☐ Domain Note
 - ☐ Definition Inherited by Attribute
 - ☐ Note Inherited by Attribute
 - ☐ Required
 - ☐ Name Inherited by Attribute
 - ☐ Domain Parent
 - ☐ Logical Datatype
 - ☐ Validation
 - ☐ Default
 - ☐ Logical Only
 - ☐ UDP Properties
- ☐  Validation Rule
- ☐  Default Value

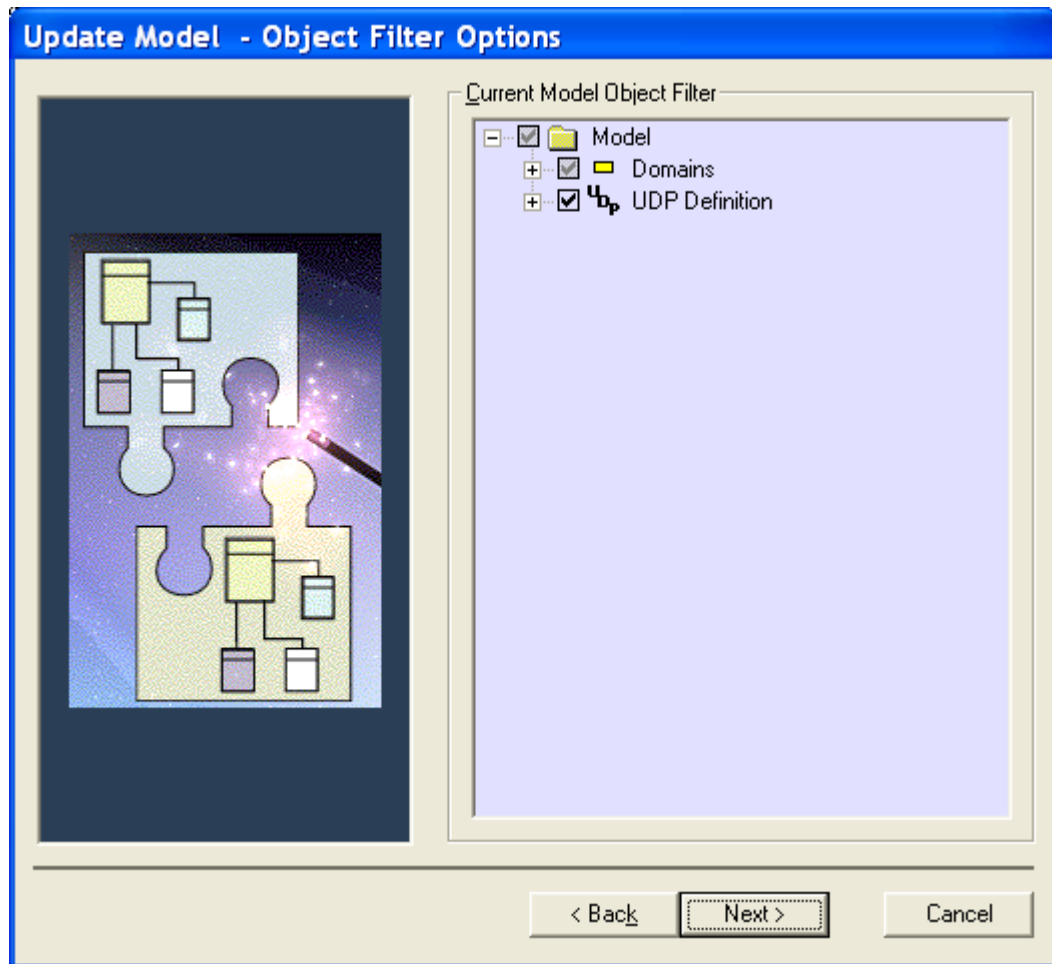
☐ Based on Subject Area(s)

< Back

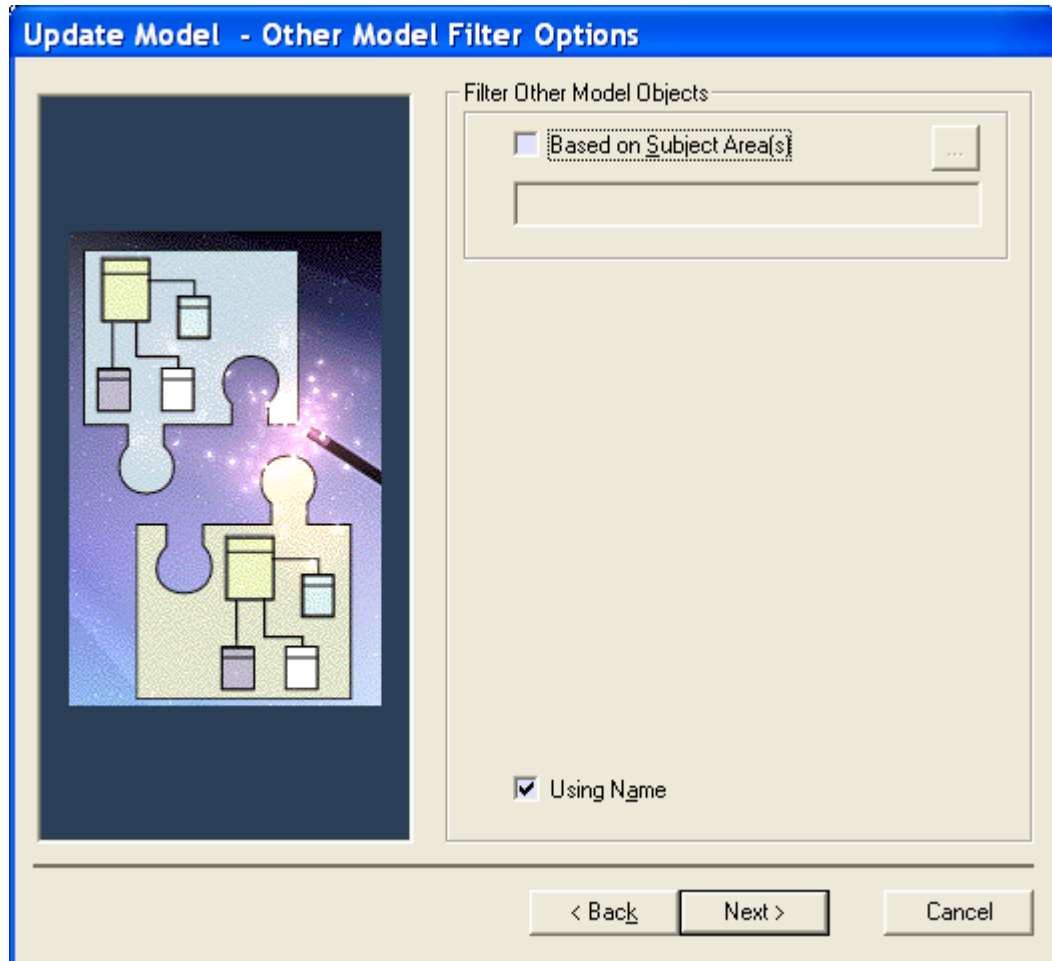
Next >

Cancel

- E. On the “Object Filter Options” screen click “Next” to accept the filters you have selected for the current model.

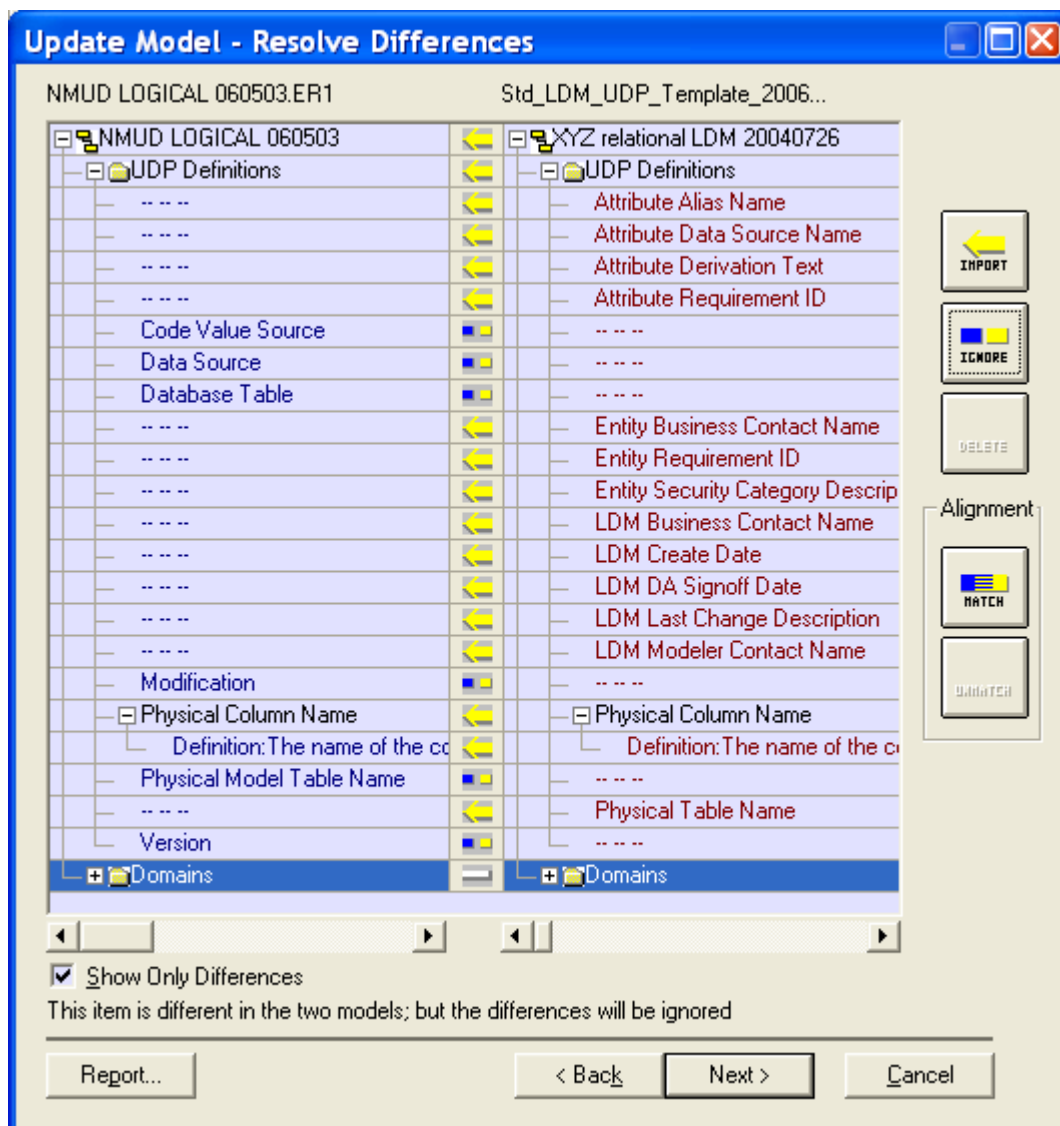


- F. On the “Other Model Filter Options” screen, first make sure that the screen looks like the figure below, and then click “Next” to compare the objects in the UDP template model with the objects you have selected for the current model. The objects will be matched on their names.



G. At the “Resolve Differences” screen you will see:

- On the left-hand side the target ERwin model into which you are going to import the UDPs. On the right-hand side the source ERwin UDP template model. There will already be a yellow arrow, pointing from right to left, just to the left of the source model template UDP objects that are to be imported into the target model.
- Scroll down the list to make sure that you did not include anything that you do not really want to import. If you see some object that you do not want to import just click on that object to highlight it. Then you can click on the blue & yellow “Ignore” button so that object will not be imported. The example in the figure below shows six UDPs and the Domains (all instances) which are marked by the tool as "Ignore". The six target model UDP names do not match any of the UDP names in the source template model.
- Checking the "Show Only Differences" box will greatly shorten the "Resolve Differences" list.



- H. Still on the “Resolve Differences” screen example you will see that the definition of the UDP Physical Column Name is marked as "Import" with the left-pointing yellow arrow. This means that the source template definition is different from the target definition.
- To inspect the differences, right click on the row with the definitions. Select "Show Details". A pop up window will show the two definitions side by side, with a blue highlight where the two items first have a difference.
 - If there is no material difference or if the meaning is essentially identical between the source and target UDPs, leave the pair marked for "Import".
 - If there is a significant difference in the meaning between the source and target UDPs, then cancel out of the compare process, rename the target model UDP to something other than the template UDP name using the UDP Dictionary, and start the compare process again.
- I. One more situation to notice on this “Resolve Differences” screen example is that there is an existing UDP named Physical Model Table Name marked as "Ignore" and a template model UDP named Physical Table Name marked as "Import"
- In order to inspect any differences in the properties of the two objects, they must be matched. First, click on the "Match" button on the right of the screen. Then select the source UDP, followed by the target UDP to be matched. Click "Yes" to confirm the resynchronization. The selected pair will now be compared and aligned in the display.
 - In this case, a "+" in the tree structure to the left of each of the UDPs just compared indicates that differences were found. Click on the "+" to expand the properties with differences. The example screen shot below shows that both the names and the definitions differ and are marked as "Import".
 - To inspect the differences, right click on the row with the names or the row with the definitions. Select "Show Details". A pop up window will show the values of the two items side by side, with a blue highlight where the two values first have a difference.
 - In this example, we know that the names are different already. We will defer acting on this difference until after we examine the difference in the definitions.
 - Using the "Show Details" feature described previously, you see that the definitions have the same intention, with the standard template UDP definition being more thorough. Making the decision to replace the existing UDP with the standard UDP, you leave both the name and the definition rows marked for "Import".
 - If there were a significant difference in the meaning between the source and target UDPs, then you would select the row with these UDPs and then click the "Unmatch" button on the right of the screen to retain the existing UDP and import the standard UDP.

Update Model - Resolve Differences

NMUD LOGICAL 060503.ER1

Std_LDM_UDP_Template_2006...

NMUD LOGICAL 060503	XYZ relational LDM 20040726
UDP Definitions	UDP Definitions
...	Attribute Alias Name
...	Attribute Data Source Name
...	Attribute Derivation Text
...	Attribute Requirement ID
Code Value Source	...
Data Source	...
Database Table	...
...	Entity Business Contact Name
...	Entity Requirement ID
...	Entity Security Category Descriptio
...	LDM Business Contact Name
...	LDM Create Date
...	LDM DA Signoff Date
...	LDM Last Change Description
...	LDM Modeler Contact Name
Modification	...
Physical Column Name	Physical Column Name
Definition: The name of the corresp	Definition: The name of the cor
Physical Model Table Name	Physical Table Name
Name: Physical Model Table Name	Name: Physical Table Name
Definition: The entity name in the ph	Definition: The name of the cor
Version	...
Domains	Domains

IMPORT

IGNORE

DELETE

Alignment

HATCH

UNHATCH

☒ Show Only Differences
 This item will be modified in The Current model

Report...

< Back

Next >

Cancel

Details of Selected Item

Current Model

Other Model

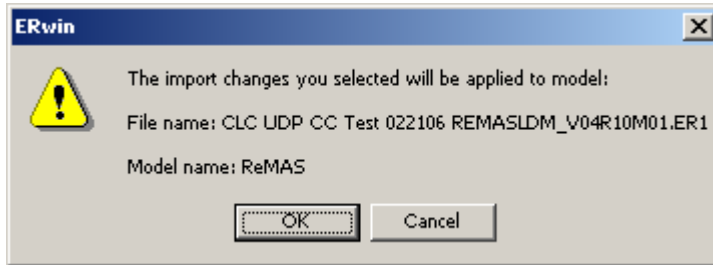
Definition: The **entity** name in the physical view of the physical model.

Definition: The **name** of the corresponding Table specified in the corresponding PDM for this LDM.
 If the physical model has not been created, this is the Table name as formed in accordance with the CMS standard naming conventions for the applicable DBMS.
 Not required for "logical only" Entities.

This item will be modified in The Current model

Close

- J. When all of the "Import" and "Ignore" decisions have been made and marked, then click “Next” at the bottom of the screen. You will see a popup screen asking you to confirm that you want to import changes and naming the target model that the changes will be imported to. Click “OK”.



K. After a brief pause you will be out of Complete Compare. By using the following procedure, you can examine the target ERwin physical model to make sure that the model contains the current standard UDPs. (The UDPs listed below are current as of late March 2006.)

- a. Select entities and then right-click on them to look at their UDP tab. You should now see four entity-level UDPs:
 - Entity Business Contact Name
 - Entity Requirement ID
 - Entity Security Category Description
 - Physical Table Name
- b. On the entities right-click and then select Attributes, You can then look at the UDP tab to see five attribute-level UDPs
 - Attribute Alias Name
 - Attribute Data Source Name
 - Attribute Derivation Text
 - Attribute Requirement ID
 - Physical Column Name
- c. At the toolbar select “Model\UDP Dictionary”. You should see five model-level UDPs:
 - LDM Business Contact Name
 - LDM Create Date
 - LDM DA Signoff Date
 - LDM Last Change Description
 - LDM Modeler Contact Name

DM G-022 Guideline for Using ERwin Complete Compare to Import Standard Physical Model UDPs

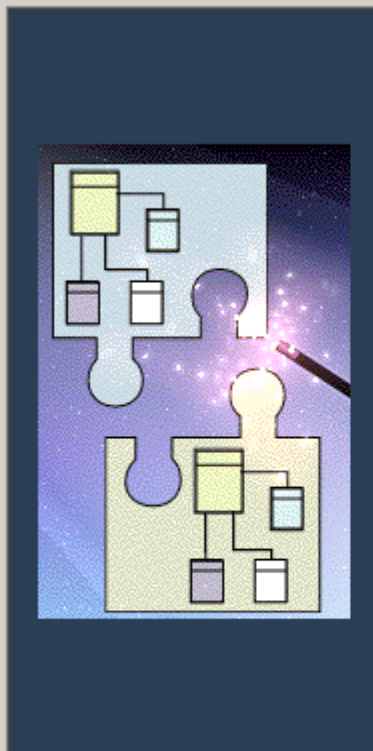
Rationale:

To manage data models as an enterprise information resource, it is necessary for the model metadata to be collected in a consistent form. The Tool Use standard defines how standard UDPs will be used as the mechanism to promote such consistency. This guideline prescribes the procedure for incorporating the standard physical data model UDPs into an existing model.

Guideline:

- A. It is imperative that you first **make a copy** of the ERwin physical data model that you wish to import UDPs into. Then immediately **rename** this copy to clearly indicate that it is the target physical data model that will be used in the import attempt. This will preserve the original physical data model in case something unwanted happens to the target physical data model during the Complete Compare work.
- B. Open up the ERwin target physical data model. Since we are talking about importing physical model UDPs, this target model will either be an ERwin “pure” physical data model, or the physical view of an ERwin logical\physical data model.
- C. From the ERwin toolbar go to “Tools\Complete Compare” and:
 - a. Set “Compare Type” to be “Model level compare”.
 - b. For the “Compare Current Model with” use the Browse feature to locate the path and file where you have saved the physical model UDP template, named “Std_PDM_UDP_Template”.
 - c. Set “Sync Action” to be “Update Current”.
 - d. Click “Next”.

Complete Compare - Set Options



Compare Options

Compare Type

- ☐ Database level compare
- ☒ Model level compare

Compare Current Model with

- ☐ Database
- ☐ ModelMart
- ☐ Script File
- ☒ ER1 File

Browse...

C:\CMS DA\CMS UDP Standards\Std_PDM_UDP

Sync Action

- ☐ Bidirectional
- ☒ Update Current
- ☐ Update Other

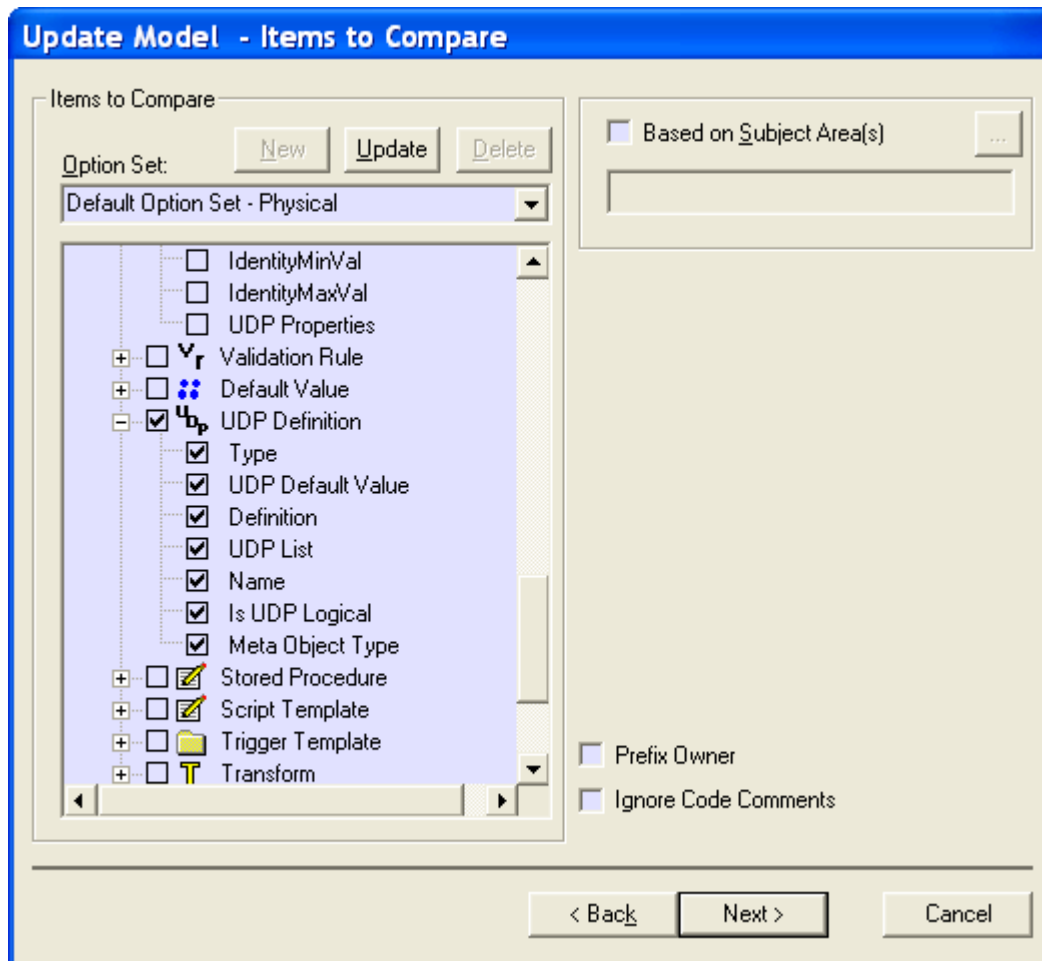
< Back

Next >

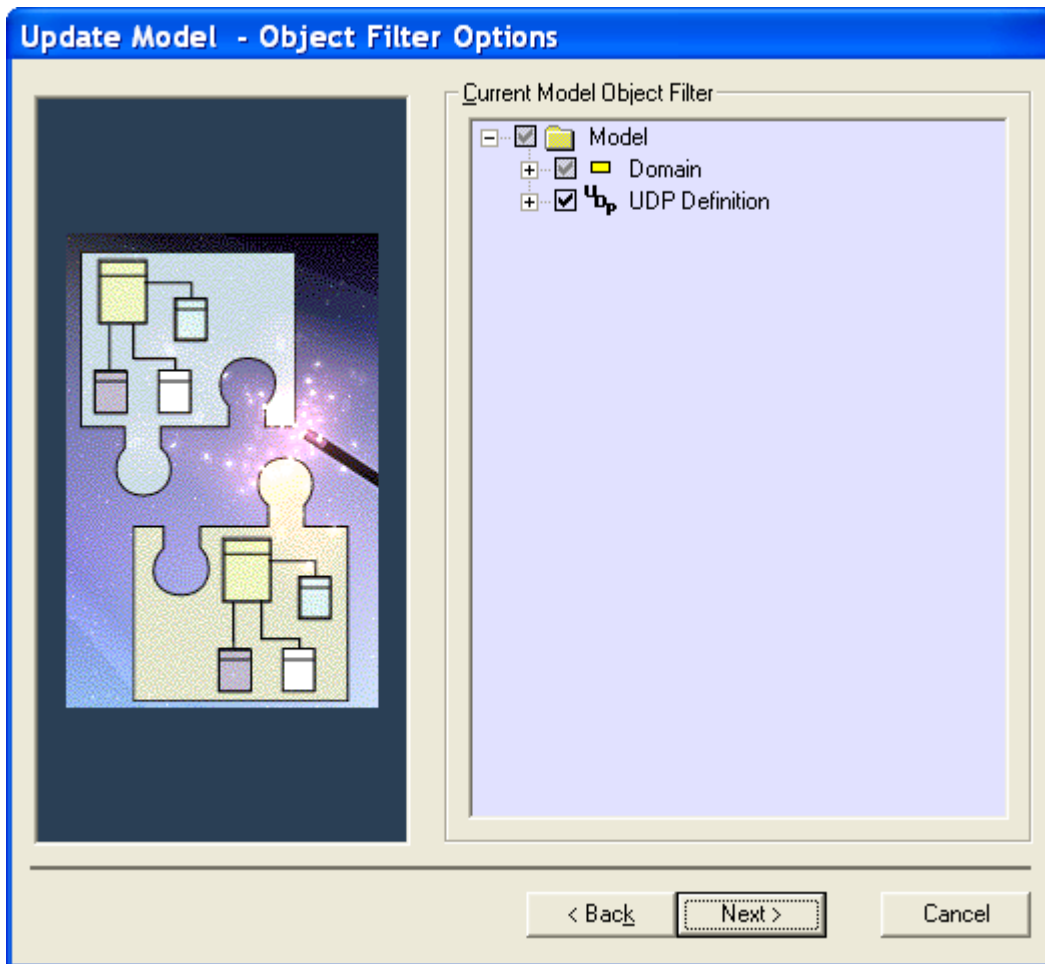
Cancel

- D. At the “Items to Compare” screen deliberately unselect everything except the UDP Definition item. All the UDP Definition subordinate items (Type, UDP Default Value and so on) definitely need to remain selected (with a checkmark next to each one). Click “Next”.

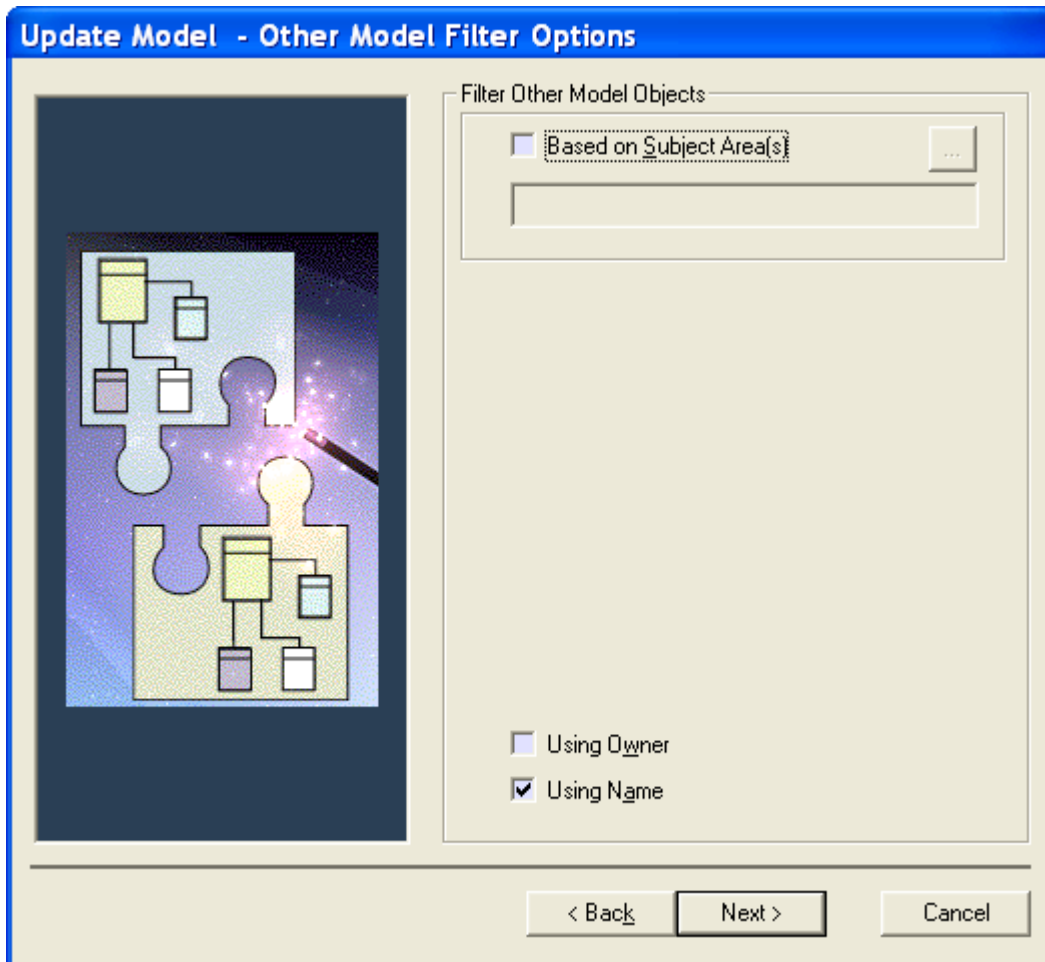
(Note that you won't be able to uncheck the Domain item – ERwin keeps that item selected. To prevent Domains from being compared, uncheck all the items subordinate to the Domain object type - Domain Name, Domain Icon and so on.)



- E. On the “Object Filter Options” screen click “Next” to accept the filters you have selected for the current model.



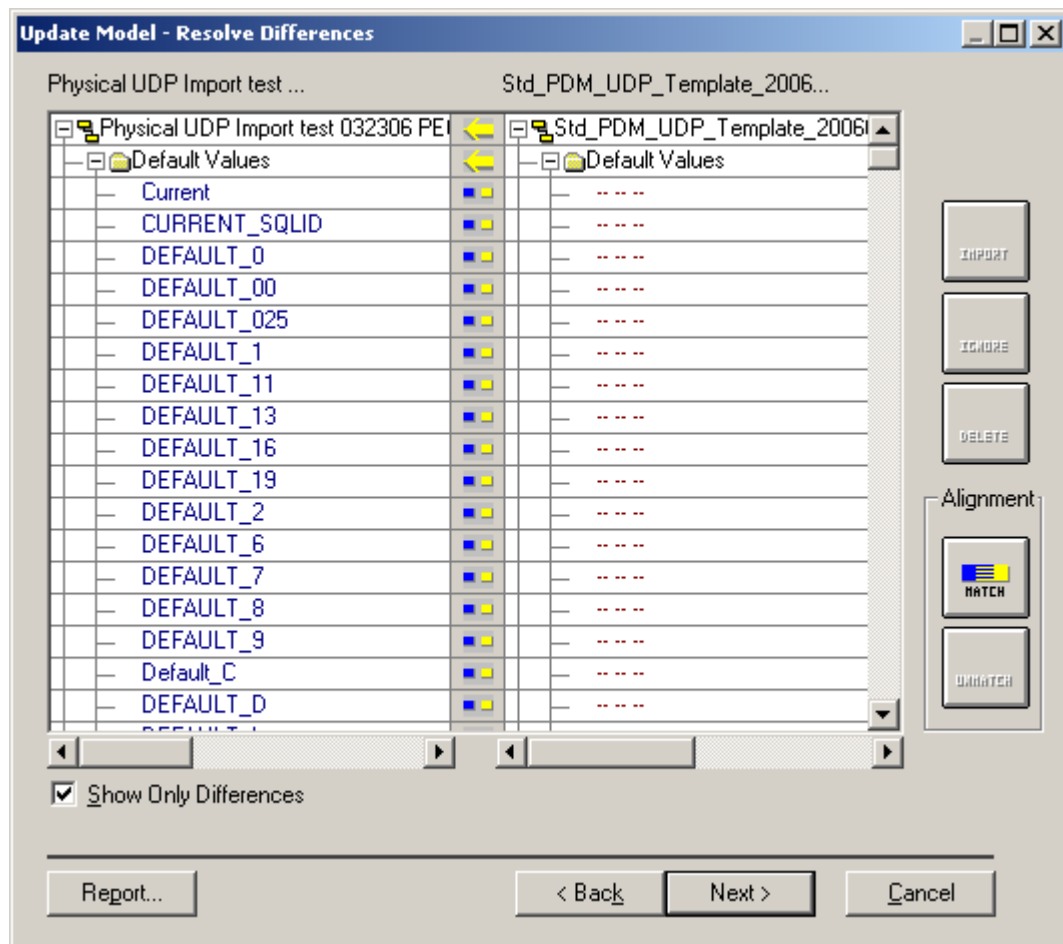
- F. On the “Other Model Filter Options” screen, first make sure that the screen looks like the figure below, and then click “Next” to compare the objects in the UDP template model with the objects you have selected for the current model. The objects will be matched on their names.



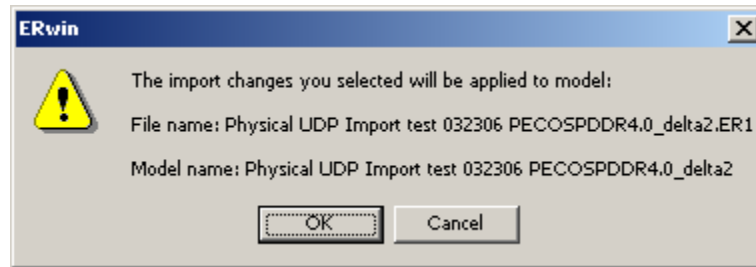
G. At the “Resolve Differences” screen you will see:

- On the left-hand side the target ERwin model into which you are going to import the UDPs.
- On the right-hand side the source ERwin UDP template model.
- There will already be a yellow arrow, pointing from right to left, just to the left of the source model template UDP objects that are to be imported into the target model.
- Scroll down the list to make sure that you did not include anything that you do not really want to import. If you see some object that you do not want to import just click on that object to highlight it. Then you can click on the blue & yellow “Ignore” button so that object will not be imported.
- Click “Next”.

Checking the "Show Only Differences" box will greatly shorten the "Resolve Differences" list.



- H. You will see a popup screen asking you to confirm that you want to import changes and naming the target model that the changes will be imported to. Click “OK”.



- I. After a brief pause you will be out of Complete Compare. By using the following procedure, you can examine the target ERwin physical model to make sure that the model contains the current standard UDPs. (The UDPs listed below are current as of late March 2006.)
- a. Select tables and then right-click on them to look at their UDP tab. You should now see two table-level UDPs:
 - Logical Entity Name
 - Table Requirement ID
 - b. On the tables right-click and then select Columns, You can then look at the UDP tab to see four column-level UDPs
 - Column Data Source Name
 - Column Derivation Text
 - Column Requirement ID
 - Logical Attribute Name
 - c. At the toolbar select “Model\UDP Dictionary”. You should see four model-level UDPs:
 - PDM Create Date
 - PDM DA Signoff Date
 - PDM Last Change Description
 - PDM System Contact Name